Identification of Driver Phone Usage Violations via State-of-the-Art Object Detection with Tracking

Steven Carrell
School of Computing
Newcastle University
Newcastle upon Tyne, United Kingdom
steven.carrell.ncl@gmail.com

Amir Atapour-Abarghouei
Department of Computer Science
Durham University
Durham, United Kingdom
amir.atapour-abarghouei@durham.ac.uk

***Abstract*—The use of mobiles phones when driving has been a major factor when it comes to road traffic incidents and the process of capturing such violations can be a laborious task. Advancements in both modern object detection frameworks and high-performance hardware has paved the way for a more automated approach when it comes to video surveillance. In this work, we propose a custom-trained state-of-the-art object detector to work with roadside cameras to capture driver phone usage without the need for human intervention. The proposed approach also addresses the issues caused by windscreen glare and introduces the steps required to remedy this. Twelve pre-trained models are fine-tuned with our custom dataset using four popular object detection methods: YOLO, SSD, Faster R-CNN, and CenterNet. Out of all the object detectors tested, YOLO yields the highest accuracy levels of up to ∼96% (AP_{10}) and frame rates of up to ∼30 FPS. DeepSORT object tracking algorithm is also integrated into the best-performing model in order to avoid logging duplicate violations.**

***Index Terms*—Mobile phone detection, Object Detection, Intelligent Transportation Systems, Deep Learning**

I. Introduction

According to the World Health Organization (WHO), approximately 1.3 million people die each year in road traffic accidents [1]. A contributing factor towards this is the use of a handheld mobile device while operating a motor vehicle, where a person is approximately four times more likely to be involved in a crash than drivers not using their phone [1].

In 2017, the UK Government doubled the penalty for using a mobile phone while driving to 6 points and a £200 fine (up from 3 points and £100) [2]. A study carried out in 2015 suggests that there is a negative correlation between a higher fine and the likelihood of a person using their phone [3]. Typically, this involves roadside police performing the laborious task of capturing the violation as it happens, likely resulting in many violations going undetected. This opens the door for a more automated process of capturing violations.

In this work, we propose a fully-automated system that will take live video from roadside surveillance cameras and detect if a driver is using a mobile phone whilst the vehicle is in operation. We will explore different quality cameras (high-end and low-end) whilst addressing challenges such as windscreen glare, tinted windows and low-light scenarios. In order for the system to be fully automated, it will need to have the ability to log each unique violation as well as to save the images corresponding to each violation.

(a) Step one. (b) Step two.

Fig. 1: Example of the proposed two-step approach: the first step (left) detects windscreen; the second step (right) first crops the driver's side and only then detects the phone.

We propose two methods for achieving this task; first, a single-step method focusing on efficiency, which will suffer a trade-off with accuracy, especially in finding extremely small objects (phone) within a larger image. The second method (two-step) focuses on achieving high accuracy by running two individually trained models simultaneously. Models used in this work include YOLO (You Only Look Once) v3 [4] and v4 [5], SSD [6], [7], Faster R-CNN [8] and Centernet [7].

Performance is measured via Average Precision (AP) [9] and the PASCAL VOC evaluation metric where the Intersection Over Union (IoU) score is >0.5 [10], as well as IoU>0.1 due to the nature of the objects being detected. The solution will be designed to work with a live video; therefore, we also evaluate the efficiency of the proposed method by calculating the frame rate of the output - i.e. frames per second (FPS). The images that are used to train our model on the phone class will predominantly be obtained and created especially for this project. In order for the model to detect mobile phone use violations with a reasonable level of accuracy, the training images will be replications of the real-world scenario of a person using their phone whilst driving (exemplar images in Figure 5). We also integrate a tracking algorithm [11] into our final model to prevent logging multiple detections for the same violation. This will allow the system to keep track of the total number of violations for a given duration.

This work aims to develop a system, working with a roadside camera 24 hours a day to detect individuals using

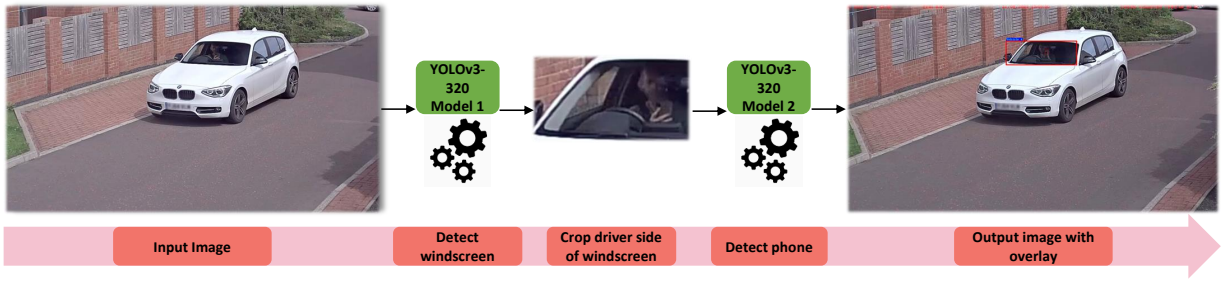


Fig. 2: Two-step approach using YOLOv3 with input size of 320×320. Each frame is resized to 320×320 and passed through the first model to detect windscreen. Image cropped on driver side is then resized to 320×320 and passed through second model to detect phone. Output image is the original frame with overlay of predicted windscreen and phone bounding boxes.

mobile phones when in operation of a motor vehicle. In short, the primary contributions of this paper are as follows:

- Train and evaluate multiple object detection methods [4]–[8] to detect mobile phone use violations with a reasonable degree of accuracy and speed.
- Test the trained models on both full images (single-step) and cropped windscreens (two-step). The single-step method operates on the full image only, while the two-step system first finds windscreen and then uses this cropped image to detect the phone (Section III).
- Establish what can be achieved on a smaller budget using a low-end consumer camera solution as well as providing insights on what is achievable with a reasonable budget using high-end cameras.
- Ensure issues such as windscreen glare and poorly-lit environments are addressed so the system can work at any time of the day.
- Integrate a tracking algorithm [11] to identify unique detections in order to log useful data whilst providing snapshots of the violations.

To better enable reproducibility, the source code for the project is publicly available[1].

II. Related Work

Recent years have seen a rise in the number of distracted driver identification systems. One such approach [12] utilises current infrastructure using LPR [13] roadside cameras and utilises a three-stage approach. The first stage is to detect the windscreen, which is then cropped and processed in the second model to identify whether a person can be clearly seen. This process is in place to ensure that the images with undesirable reflection effects are not processed. The final stage will then detect for mobile phone usage [12]. Images could not be acquired during summer days between 12:00 and 15:00 due to excessive amount of windscreen glare [12]. Initial testing for this work shows windscreen glare for the majority of the day (as discussed in Section III-A1), suggesting that this would not be appropriate for some territories.

Another study builds upon a software already developed by the Dutch Police, which first looks for a Dutch licence plate and then outputs the driver's side of the windscreen [14]. These images are used as inputs to the trained model where hands, phone and face are detected. The model looks to eliminate the issue of falsely classifying objects such as phone mounts by checking where the phone is positioned in relation to the head and hand; if the distance is greater than a set threshold, then it is not classified. The system is dependent on the Dutch Police windscreen detector, which only works for Dutch Plates. Due to the reliance on a third party software to make this work, there is a lack of control on a significant portion of the overall approach. There could be issues with support further down the line, or changes to licensing.

Another work monitors the driver using their phone in addition to hand position [15]. This approach uses cameras inside the vehicle positioned towards both the driver and the steering wheel to detect both mobile phone and hands [8]. Geometric information is then extracted to determine if the driver is using their phone [15]. Mass deployment of this system could prove costly and impractical due to its reliance on cameras within the vehicle.

Our proposed system looks to solve the limitations of prior work starting with the issues of windscreen glare where the use of a polarizing filter on the camera lens has been explored. Additionally, our work addresses the issue of identifying unique violations - neglected in prior work - by implementing a tracking algorithm [11]. Our approach utilises a standard video surveillance roadside camera, which limits the cost of deployment. The next section describes our approach for building this system in more detail.

III. Approach

Here, we describe the steps taken to build the proposed solution of a fully-automated system to detect driver violations. We propose two methods for achieving this: a single-step and a two-step approach (Figure 2), where by step we refer to a dedicated trained model in the overall system architecture. The single-step model is trained to detect both a licence plate and a person using their phone from a single image input in one forward pass through the model. A key advantage of this approach is that running a single model to complete the entire task at once results in a more lightweight faster system. A potential limitation of this method, however, would be the trade-off with accuracy as the trained model will be attempting to detect a very small object within a large image.

[1] https://github.com/carrell-ncl/Windscreen2

(a) Low-end camera (b) High-end camera

Fig. 3: Images captured using high-end and low-end cameras.

Equipment	Make Origin	Model number	Spec
Camera	Avigilon/Canada	2.0C-H5A-B1	2MP
Camera	Axis/Sweden	P1353	1.3MP
Camera	ELP/China	ELP-USB-FHD01M-SFV	2MP
Infra-Red	Raytec/UK	VAR2-i8-1	850nm
Infra-Red	Raytec/UK	VAR2-i8-1-730	730nm

TABLE I: List of cameras and IR used for testing.

To remedy this issue, we also propose a two-step solution, which first detects the windscreen of the vehicle, and then uses the cropped image of only the driver side as the input for the next step to search for the phone. An overview of the process of the two-step approach is seen in Figure 2.

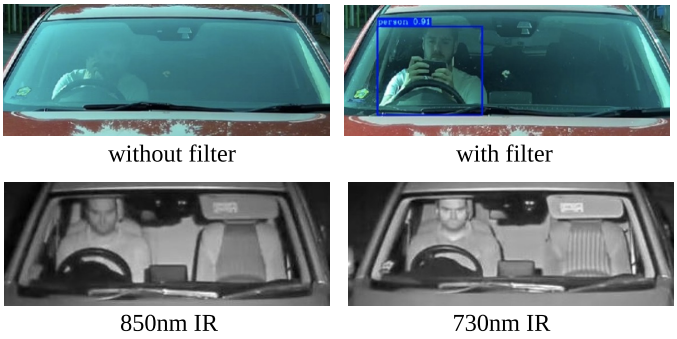
We evaluate four popular frameworks with various backbone models and image input sizes [3], [5], [6], [8]. In total, we fine-tune 12 pre-trained models with our custom dataset, where we have phone and licence plate for the single-step method, and windscreen and phone for the two-step one.

To train and evaluate the models, images are acquired using high-end (Aviglon, Axis) and low-end (ELP) cameras under varying weather conditions. Details of all equipment used can be found in Table I. Figure 3 demonstrates the difference in quality between a high and low-end camera, where all other conditions are identical. We discuss the camera and hardware setup in the following section.

A. Camera and hardware setup

A practical feasibility study is carried out to ensure that images of a high enough quality could be captured through a car windscreen in all weather conditions. As pointed out in [12], images captured in certain hours of the day present the challenge of windscreen glare, whilst night-time and poorly-lit areas can result in dark unusable images. We also acknowledge that tinted windscreens may result in cameras not having reasonable visibility into the vehicle. While this can simply be resolved by having the camera in night-mode and using excessive amounts of IR, this issue is not common in the UK due to legal restrictions, so it is beyond the scope of this work.

The primary objective of this paper is to not just create an object detector that could capture violations, but one that could do this during all hours of the day. In this section, we address the following challenges and propose solutions.



without filter with filter

850nm IR 730nm IR

Fig. 4: Top: images captured with and without a polarizing filter. Note the successful detection with filter. Bottom: Night images with active IR - 850nm vs 730nm

1) Windscreen Glare: One of the most difficult challenges in seeing inside the vehicle is windscreen glare, which usually occurs when the sun is in a particular part of the sky and can be made worse with clouds as they can be reflected on the windscreen. Our preliminary tests found that the issue of glare would occur during the majority of the day.

We can resolve this using a polarizing filter which is fixed to the camera lens. The effectiveness of this solution is demonstrated in Figure 4 (top), which shows the same image taken with and without a polarizing filter. We tested the impact of the polarizing filter using a pre-trained YOLOv3 model to see if it can detect the person inside the vehicle. As seen in Figure 4, it cannot detect the person without a filter but detects the person with 91% confidence when the polarizing filter is added, which points to the importance of the polarizing filter.

2) Low-light conditions: In order for the system to function successfully in low-light conditions, we would need to consider an appropriate light source. Directional white visible light cannot be used as it causes glare to the driver. Instead, active Infra Red (IR) is used, where two different wavelengths are tested: 850 and 730 nanometers (nm). Figure 4 (bottom) demonstrates both wavelengths of IR working well in low-light conditions. However, it is clear that the camera is able to capture more details using 730nm. Details of the IR used for this work can be found in Table I. The deployed system will be designed to expect both RGB (day) and monochrome images (night). A proportion of monochrome images taken with IR have been used in the training and testing of the model.

B. Dataset

This section describes the data used for training and testing.

1) Training images: For the single-step approach, the dataset consists of 2,150 images of phones and 2,235 images of licence plates. The licence plate images are obtained from the Google Open Images Dataset [16]. For the phone class, the main proportion of images are obtained specifically for this project with a mixture of quality and weather conditions to best represent the real-world scenarios (Figure 5). For the phone detections, the majority of what is actually being detected is multiple variations of hand positions holding a phone. To address this, 80% of images are collected specifically for this

Fig. 5: Bespoke RGB and monochrome training images of varying quality obtained specifically for this work.

application under varying conditions (Figure 5). The two-step approach is trained using 487 images of windscreens to train the first model, which will be used for cropping. The second model is trained using only the phone images.

2) Test images: The images used to evaluate the trained models consist of 216 images of a person using their phone whilst driving. Since public traffic camera footage is not readily available, test images were captured with the aid of volunteers to best represent a real-world scenario. Images were obtained using a mixture of high-end and low-end cameras (Table I) and separate to those taken for training of the models. Test images for the two-step approach were obtained by cropping out only the windscreen of these same images.

C. Training the object detectors

One of the main challenges for successfully detecting a person using their phone is the ability for the system to detect small objects with significant variation. To address this, multiple object detection methods are trained and evaluated. The same trained models are used to evaluate both the single-step and two-step approaches.

A variety of pre-trained base networks (backbones) are chosen depending on the object detector used (Table II). All these base networks with the exception of MobileNet, are typically used when running on a GPU platform [6]. Here, we include MobileNet as one of our trained models to see how effective a low-cost light-weight detector is. An application such as this may benefit from running on an edge device [17] where a more light-weight model optimised for smaller computational resources would be preferred.

1) YOLOv3 and YOLOv4: Results from other studies [4], [5] have concluded that higher-resolution models yield greater accuracy whilst lower-resolution models run at a higher frame rate. We thus opt for training both YOLOv3 and YOLOv4 with input resolutions of 512×512, 416×416 and 320×320.

2) Faster R-CNN, SSD, and Centernet: The remaining frameworks are obtained and fine-tuned using the TensorFlow Object Detection API [18], [19], which provides different pre-trained models that could be fine-tuned on our custom dataset.

D. Evaluating the models

To replicate a real-world scenario, the majority of the test images are obtained at a distance of 20-30m from the camera, with a height of approximately 3m. These images are captured during different times of the day under varying weather conditions to enable testing the generalisation capabilities of the system as well as its predictive performance.

Object detector	full		cropped		detection
	AP_{50}	AP_{10}	AP_{50}	AP_{10}	FPS
YOLOv4 512	**45.98**	**73.11**	59.62	83.32	27.12
YOLOv3 512	35.81	46.16	**63.27**	85.88	25.18
YOLOv4 416	40.23	48.23	61.60	**87.58**	26.12
YOLOv3 416	37.41	51.64	58.44	79.99	25.96
YOLOv4 320	19.62	52.36	52.65	81.93	26.05
YOLOv3 320	37.54	72.45	59.05	84.62	28.97
C-Res101 512	43.81	56.18	50.04	66.60	35.90
F-Res101 640	37.18	48.57	41.78	55.50	15.40
F-Res152 640	34.85	44.48	40.89	50.57	11.50
S-M FPNLite 640	12.23	18.23	12.64	28.41	**42.77**
S-R50 FPN 640	2.77	7.77	0.92	4.22	23.66
S-R101 FPN 640	0.60	4.86	0.71	13.90	22.20

TABLE II: Results showing average precision and frame rate. C-Res denotes Centernet ResNet, F-Res represents Faster R-CNN ResNet, S-M refers to SSD Mobilenet and S-R is SSD ResNet. Cropped refers to images of the windscreen for the two-step approach. FPS is based on detection only and does not include tracking and the two-step approach.

Test images for evaluation are split into 2 sets, the first using 216 full image snapshots taken from the camera, the next with the same images but with only the cropped windscreen. This allows us to determine if the model will perform more favourably with the single-step or the two-step system.

Models are evaluated using average precision with IoU thresholds of >0.5 and >0.1 hereby referred to as AP_{50} and AP_{10}. We attempt to find very small objects (phone) with significant variation, so the lower IoU threshold is more appropriate. Frames per second (FPS) is another metric used to evaluate the speed of our system. We run the same test video for each model and then calculate the average FPS.

1) Choosing the best model: The test images are split into two categories, namely high-quality and low-quality. High-quality images are captured using the high-end cameras (Avigilon and Axis) to represent how the model should perform when the system has been built with a relatively larger budget in mind. The low-quality images are acquired using the low-end camera (ELP) to demonstrate how the model will perform under cost-effective considerations. Details of the cameras are listed in Table I. The speed in which these models perform also needs to be re-evaluated to incorporate the two-step windscreen method as well as the object tracking algorithm.

2) Object tracking and data collection: A consideration when building a fully-automated system is how the phone violations are going to be recorded in a way that is useful to the end user. To do this, the system would have to be able to distinguish between unique and duplicate detections. For example, a five second video may show one driver using their phone, but since the detections are done per frame, it may count duplicate violations for every one of these frames. In order to address this, we add DeepSORT [11], which is an object tracking algorithm. This will add a unique ID for each detection and then takes each frame to predict if the next detection belongs to the same ID or not.

Fig. 6: False positive result for AP_{50}, which demonstrates that for the application in this work, AP_{10} is more appropriate.

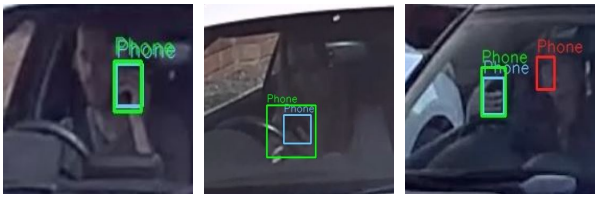


Fig. 7: Results from YOLOv3 320 with AP_{10} showing high accuracy but high false positives on low-quality images. Blue bounding boxes denote ground truth, green refers to the true positive prediction and red is the false positive prediction.

IV. Experimental Results

In this section, we evaluate our models using the experimental setup and the metrics discussed in the previous section.

A. System Specification

TensorFlow 2.2 with an Nvidia RTX2080Ti GPU on a Windows 10 system with an AMD Ryzen 7 3800X CPU and 32 GB of memory is used for all implementation and testing.

B. Average precision

For the first accuracy test, we present the single-step method where the full test image is used in the model trained to detect the phone. Table II shows the YOLO models outperforming the other object detectors. YOLOv4 with input size 512 is best performing on both AP_{50} and AP_{10}, whilst the SSD models are the poorest performing for both IOU thresholds.

Next, we test the same models, but this time with the cropped windscreen images to determine whether the two-step approach is more appropriate. The results (Table II) suggest that if accuracy was the main driver, the two-step method will be more favourable to use, giving higher AP in almost all of the trained object detectors. Again, the YOLO models yield the highest accuracy scores. YOLOv3 with the larger input size gives the best results with AP_{50}, whilst YOLOv4 with the input resolution of 416 gives the highest accuracy for AP_{10}.

Review of the predictions made on the test images when IOU threshold is set to $>$0.5 shows multiple false positives despite the predicted bounding box surrounding the correct object. As mentioned previously, the objects that the model is attempting to predict have significant amounts of variation, meaning that it will always be difficult to get a high IOU score. Figure 6 shows a false positive prediction where we have the IOU threshold set to $>$0.5. We can see that the model is capturing the violation correctly, but narrowly missing the IOU threshold resulting in a false positive. Based on this, we propose an IOU threshold of $>$0.1 for this application.

C. Frame rate

Speed of the trained models is evaluated using a 70-second test video, calculated on detection only (prior to the addition of tracking and the two-step method). Table II shows the single-stage (YOLO, SSD, CenterNet) object detectors are significantly faster than the two-stage R-CNN detectors. As expected, the most efficient object detector is the low-cost SSD Mobilenet FPNLite 640 with 43 FPS, however lacking in accuracy. Each of the YOLO models seem to perform consistently well with regards to both accuracy and speed with YOLOv3 320 performing the best at 29 FPS.

D. Output images

Figure 7 demonstrates sample results obtained from the YOLOv3-320 model for the two-step method; blue bounding box refers to the ground truth, green is the true positive prediction, and red is the false positive prediction. Although the model performs well on the low-quality camera, there appears to be a higher chance of a false positive.

Having a high proportion of false positives for this particular application could result in incorrectly fining individuals. It may, therefore, be appropriate to increase the score threshold for the phone detector step to reduce these false positives. Observation from the predictions of the test set confirm that AP_{10} is appropriate for this application of detecting such small and difficult images, as demonstrated in Figure 6.

The chosen model is YOLOv3 with the input size of 320 using the two-step method. Although it did not achieve the highest overall accuracy, it came a close third behind YOLOv3-512 and YOLOv4-416 (Table II). The deciding factor is the speed of the model, as it is able to achieve almost 29 FPS (almost 11% faster then the next model). For this type of application, the cameras typically monitor fast-moving traffic. Having a model with a smaller input size means it will be less expensive with regards to hardware demands.

E. Integrating the two-step method and tracking

The next stage in our overall system is to re-train the chosen YOLOv3-320 model and modify the code for the two-step approach, where the first step is trained to detect the windscreen and the next step trained on only the phone images. The DeepSORT [11] tracking algorithm is also integrated into this system. The frame rate is recalculated on the same video to show the impact of the tracking algorithm and the two-step approach on system efficiency. Tracking algorithm reduces FPS by almost 10%, whilst adding the extra step on top of this sees a further reduction of $\sim$50% giving a frame rate of 13.15 FPS on the YOLOv3-320 model.

For the final benchmark tests, we split the test images into 2 categories; high-quality and low-quality, with 116 and 100 images respectively. Based on our chosen metric of IoU threshold greater than 0.1, for YOLOv3-320, we can achieve an AP of as high as 95.81% on the images taken with only

Detector	Method	HE		LE	
		AP_{50}	AP_{10}	AP_{50}	AP_{10}
YOLOv3 320	Two-step	78.29	95.81	41.93	74.36

TABLE III: High-end (HE) and low-end (LE) camera results.

high-quality cameras, whilst still achieving an AP of 74.36% on images taken from the low-quality camera (Table III).

V. Discussions and Future Work

Our proposed approach delivers very promising results and further enables a fully-automated end-to-end surveillance system capable of capturing mobile use violations while driving. However, there are still limitations that need to be addressed before tangible impact can be made.

The proposed two-step model detects the driver side of the windscreen based on right-hand drive vehicles. When deployed in countries using left-hand drive vehicles, we can simply crop the opposite side. Alternatively, even this process can be automated by detecting the licence plate to identify country and determine which side is the driver.

Section III-B addresses not having access to public roadside cameras, so the next step would be to deploy the system with support from local authority/police. Test parameters of this work have been based on 3m mounting height with a distance of 25-30m from the subject, meaning that phone could be hidden when texting close to lap. This could be remedied when deployed on a public road by utilizing a gantry traffic camera which allows for a better view within the vehicle.

Although, the two-stage approach achieves greater accuracy, there is a compromise with frame rate. We propose optimizing the model and exploring TensorRT [20] framework to potentially improve the speed of the model.

VI. Conclusion

In this paper, we have presented a deep learning approach for detecting driver phone violations in all weather conditions without the need for human intervention. A total of 12 object detection models [3], [5]–[8], [21], [22] are fine-tuned and evaluated based on speed and accuracy for both the approaches which are: single-step, where a single frame is used to detect the phone, and the two-step, which first detects windscreen and then uses the cropped image of only the driver side to detect the phone. The two-step approach yields higher accuracy but lower frame rate due to having to run two models simultaneously. The model chosen based on both accuracy and speed is YOLOv3 with an input resolution of 320. We also integrate DeepSORT, an object tracking algorithm, which allows us to only collect and log unique phone detections from the driver's side, meaning that this collected data could be made useful for time-series analysis. The object detector is able to achieve an accuracy of 84.62% (AP_{10}) on the 216 test images, with a frame rate of 13.15 FPS during activity and ∼27 with no activity. For high-end camera and low-end camera experiments, we achieve accuracy levels as high as 95.81% on the high-end and 74.36% on the budget cameras.

We kindly invite the readers to refer to the supplemental ***video****: https://youtu.be/PErIUr3Cxvg for more information and more results in video format.*

References

[1] World Health Organization. Road traffic injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, 2021. [Online; accessed 5-July-2021].

[2] GOV UK. Double penalties for motorists using mobiles. https://www.gov.uk/government/news/double-penalties-for-motorists-using-mobiles, 2017. [Online; accessed 7-July-2021].

[3] Luis Márquez, Víctor Cantillo, and Julián Arellana. Mobile phone use while driving: A hybrid modeling approach. *Accident Analysis & Prevention*, 78:73–80, 2015.

[4] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[6] Yixuan Kang. Research on ssd base network. In *IOP Conference Series: Materials Science and Engineering*, volume 768, page 072031. IOP Publishing, 2020.

[7] Zhujun Xu, Emir Hrustic, and Damien Vivet. Centernet heatmap propagation for real-time video object detection. In *European Conference on Computer Vision*, pages 220–234. Springer, 2020.

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[10] Paul Henderson and Vittorio Ferrari. End-to-end training of object class detectors for mean average precision. In *Asian Conference on Computer Vision*, pages 198–213. Springer, 2016.

[11] Xinyu Hou, Yi Wang, and Lap-Pui Chau. Vehicle tracking using deep sort with low confidence track filtering. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2019.

[12] Bensu Alkan, Burak Balci, Alperen Elihos, and Yusuf Artan. Driver cell phone usage violation detection using license plate recognition camera images. In *VEHITS*, pages 468–474, 2019.

[13] Clive Norris and Xavier L'Hoiry. Times of crises and the development of the police national automatic number plate recognition system in the uk. In *Big Data, Surveillance and Crisis Management*, pages 198–221. Routledge, 2017.

[14] Jannes W Elings. Driver handheld cell phone usage detection. Master's thesis, 2018.

[15] T Hoang Ngan Le, Yutong Zheng, Chenchen Zhu, Khoa Luu, and Marios Savvides. Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 46–53, 2016.

[16] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[17] Beatriz Blanco-Filgueira, Daniel Garcia-Lesta, Mauro Fernández-Sanjurjo, Víctor Manuel Brea, and Paula López. Deep learning-based multiple object visual tracking on embedded system for iot and mobile edge computing applications. *IEEE Internet of Things Journal*, 6(3):5423–5431, 2019.

[18] Pirkko Mustamo. Object detection in sports: Tensorflow object detection api case study. *University of Oulu*, 2018.

[19] Tensorflow. Tensorflow 2 detection model zoo. https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md, 2021. [Online; accessed 5-May-2021].

[20] Han Vanholder. Efficient inference with tensorrt, 2016.

[21] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019.

[22] Wei Wang, Yutao Li, Ting Zou, Xin Wang, Jieyu You, and Yanhong Luo. A novel image classification approach via dense-mobilenet models. *Mobile Information Systems*, 2020, 2020.